CEWES MSRC/PET TR/98-46

Evaluating New Transparent Persistence Commodity Models: JDBC, CORBA PSS, OLEDB and W3C WOM for HPC T&E Databases

by

Geoffrey C. Fox
W. Furmanski
T. Pulikal

08h00498

Work funded wholly or in part by the DoD High Performance Computing Modernization Program CEWES Major Shared Resource Center through

Programming Environment and Training (PET)

Supported by Contract Number: DAHC 94-96-C0002
Nichols Research

Evaluating New Transparent Persistence Commodity Models: JDBC, CORBA PSS, OLEDB and W3C WOM for HPC T&E Databases

G.C. Fox, W. Furmanski and T. Pulikal
Northeast Parallel Architectures Center, Syracuse University, Syracuse NY 13244-4100
gcf@npac.syr.edu, furm@npac.syr.edu

Abstract

We analyze here the standard candidates for the universal (storage medium- and vendor-independent) persistency frameworks as proposed by the leading alternative technologies for distributed objects: Java, CORBA, COM and WOM. We point out that the consensus in this field is yet to be reached and we present our Pragmatic Object Web approach to coordinate and integrate complementary technologies. We illustrate it on a few practical examples of Web linked collaboratory database environments we constructed recently for various communities and application domains such as telemedicine, distance education, interactive FMS training, data mining of T&E data from the Virtual Proving Ground. Finally, we summarize lessons learned and we outline our recommendations for the HPC T&E database approach.

Introduction

The distributed and voluminous data produced during various testing and evaluation procedures needs to be stored and managed in an efficient, reliable and scalable manner. The data produced could be on diverse platforms, stored in various formats, and accessed using different methods. Update rates can widely vary, ranging from static, read-only historical databases accumulated in a data warehouse to dynamic 'datafoam' acting as transient database storage for real-time interactive simulations. Front-end requirements for advanced T&E applications such as Virtual Prototyping are also diverse and typically more demanding and graphics / visualization oriented than the traditional client-server transaction processing applications.

Recently, a significant progress has been made by the Web / Commodity Computing in the area of Web linked databases. The current systems evolve from the client-server towards the 3-tier *Object Web* [1] models with the distributed object / componentware middleware responsible for the 'business / application logic' inserted between the Web browser based interactive front-ends and the backend databases. Such 3-tier organization provides us also with a natural software bus framework to plug-and-play specialized high performance computing modules.

We are currently exploring various strategies for integrating the storage and management of large, heterogeneous and distributed datastores, in a simple, location- and datastore- transparent way. For this purpose, we are analyzing, evaluating and comparing the following four major technologies competing as candidates for distributed object/componentware computing standards: Java by Sun Microsystems, CORBA by Object Management Group, COM by Microsoft and XML/RDF/DOM (sometimes referred to as WOM [2]) by the World-Wide-Web Consortium.

In our Pragmatic Object Web [3] approach at NPAC we adopt the integrative methodology i.e. we setup a multiple-standards based framework in which the best assets of various approaches accumulate and cooperate rather than competing. We implement it by building a Java server (JWORB) [4] which handles multiple network protocols and includes currently support both for HTTP (Web) and IIOP (CORBA), and soon also for DCE RPC (DCOM) and RMI (Pure Java). A

mesh of collaborating JWORB servers forms a Pragmatic Object Web middleware and it connects naturally to Web browsers in the front-end and to legacy systems such as databases or HPC modules in the backend, using one of the supported standard protocols.

In a set of PET FMS tasks within the DoD High Performance Modernization Program, we are adapting our generic POW technologies such as JWORB to the DoD needs in the form of an emergent WebHLA framework that we believe is capable to integrate High Performance Modeling and Simulation with the advanced computing requirements of the Test and Evaluation community. Our four papers submitted to this conference address various aspects of WebHLA such as: Web/Commodity based front-ends [5]; JWORB based RTI software bus in the middleware [6]; POW based database backends (this paper); and WebFlow based visual authoring and integration across all tiers [7].

In this paper, we focus on the backend database layer and we first summarize and evaluate transparent persistence models as currently offered by Java , CORBA, COM and WOM. Next, we illustrate our current approach in the context of a few selected collaborative database applications such as CareWeb (telemedicine), Language Connect University (distance education), FMS Training Space (interactive training), and we also report on our early experiments in Data Mining for a Virtual Proving Ground database. Finally, we summarize the current status in the field and our suggested strategy towards FMS and T&E integration within the WebHLA framework.

Persistence Models on Pragmatic Object Web

We summarize here the ongoing activities within the Java, CORBA, COM and WOM communities in the area of universal persistence frameworks, i.e. abstract data models that would span multiple vendors and various storage media such as relational or object databases and flat file systems.

JDBC and JavaBlend JavaSoft's Java Database Connectivity (JDBC) is a standard SQL database access interface for accessing a wide range of relational databases from Java programs. It encapsulates the various DBMS vendor proprietary protocols and database operations and enables applications to use a single high level API for homogenous data access. JDBC API defined as a set of classes and interfaces supports multiple connections to different databases simultaneously.

JDBC API mainly consist of classes and interfaces representing database connections, SQL statements, result sets, database metadata, driver management etc. The main strength behind the JDBC API is its platform- and database-independence and ease of use, combined with the powerful set of database capabilities to build sophisticated database applications. The new JDBC specification (JDBC 2.0) which was released recently adds more functionality like support for forward and backward scrolling, batch updates and advanced data types like BLOB, and Rowsets which are JavaBeans that could be used in any JavaBean component development, etc. Other important features include support for connection pooling, distributed transaction support and better support for storing Java objects in the database.

Despite of the simplicity of use and the wide acceptability, the JDBC API has its own disadvantages. The API is primarily designed for relational database management systems and thus is not ideal for use with object databases or other non-SQL databases. Also, there are several problems due to various inconsistencies present in the driver implementations currently available.

JavaBlend, a high-level database development tool from JavaSoft that will be released this year, enables enterprises to simplify database application development and offers increased performance, sophisticated caching algorithms and query processing to offload bottlenecked database servers. It is highly scalable because it is multi-threaded and has built-in concurrency control mechanisms. It provides a good object-relational mapping that fits best into the Java object model.

UDA: OLEDB and ADO Universal Data Access (UDA) is Microsoft's strategy for high performance data access to a variety of information sources ranging from relational to object databases to flat file systems. UDA is based on open industry standards collectively called the Microsoft Data Access Components. OLEDB, which is the core of Universal Data Access strategy, defines a set of COM interfaces for exposing, consuming and processing of data. OLEDB consists of three types of components: *data providers* that expose or contain data; *data consumers* that use this data; and *service components* that processes or transforms this data. The data provider is the storage-specific layer that exposes the data present in various data stores like relational DBMS, ORDBMS, flat files for use by the data consumer via the universal (store-independent) API.

OLEDB consists of several core COM such as *Enumerators, Data Sources, Commands* and *Rowsets*. Apart from these components, OLEDB also exposes interfaces for catalog information or metadata information about the database and supports event notifications by which consumers sharing a rowset could be notified of changes at real time. Other features of OLEDB that will be added in future are interfaces to support authentication, authorization and administration as well as interfaces for distributed and transparent data access across threads, process and machine boundaries.

While OLEDB is Microsoft's system-level programming interface to diverse data sources, ActiveX Data Objects (ADO) offers a popular, high / application-level data consumer interface to diverse data. The main benefits of ADO are the ease of use, high speed, low memory overhead, language independence and other benefits that comes with the client side data caching and manipulation. Since ADO is a high-level programming interface application developers need not be concerned about memory management and other low-level operations. Some of the main objects in ADO Object model are: *Connection, Command, Recordset, Property, Field.*

CORBA Persistent State Service The initial CORBA standard that was accepted by OMG in the persistent objects domain was the Persistent Object Services (POS). The main goals for such a service were to support corporate centric datastores and to provide a datastore independent and open architecture framework that allows new datastore products to be plugged in at any time. POS consisted of four main interfaces namely, the Persistent Object interface (PO) that the clients would implement, Persistent Idinterface (PID) for identifying the PO object, Persistent Object Manager interface (POM) that manages the POS objects and the Persistent Data Service interface(PDS) which actually does the communication with the datastore.

Although this specification was adopted more than three years ago, it saw very little implementations because of it's complexity and inconsistencies. The specification also exposed persistence notion to CORBA clients which was not desirable and the integration with other CORBA services were not well defined. Thus OMG issued a new request for proposal for a new specification, *Persistent Object State (PSS)* that is much simpler to use and to implement and is readily applicable to existing data stores.

The Persistent State Service specification, currently still at the level of an evolving proposal to OMG led by Iona / Orbix, uses the value notation defined in the new Objects by Value specification for representing the state of the mobile objects. The PSS provides a service to object implementors which is transparent to a client. This specification focuses on how the CORBA objects interact with the datastore through an internal interface not exposed to the client. The persistent-values are implemented by application developers and are specific to a datastore and can make use of the features of the datastore. The specification also defines interfaces for application-independent features like transaction management and association of CORBA objects with persistent-values.

Web Object Model World-Wide Web Consortium (W3C) develops a suite of new Web data representation and/or description standards such as XML (eXtensible Markup Language), DOM (Document Object Model) or RDF (Resource Description Framework). Each of these technologies has a merit in its own but when combined they can be viewed collectively as a new, very dynamic, flexible and powerful *Web Object Model* (WOM) [2].

XML is a subset of SGML that acts as a metamodel for specialized markup languages i.e. it allows to define new custom / domain specific tags and document templates or DTDs (Document Type Definitions). Such DTDs provide a natural bridge between Web and Object Technologies since XML documents can be now viewed as instances of the associated DTD classes. DOM makes such analogy even more explicit by offering an orthodox object-oriented API (specified as CORBA IDL) to XML documents. Finally, RDF offers a metadata framework that allows to associate a set of named properties and their values with a particular Web resource (URL). In the WOM context, RDF is used to bind in a dynmaic and transient fashion the Web Object methods located in some programming langauge files with the Web Object states, specified in XML files.

Summary As seen from the above discussion, the universal data models are just emerging. Even if most major RDBMS vendors are OMG members, the consensus in the CORBA database community is yet to be reached. WOM is a new, '97 / '98 concept and several aspects of and relations between the WOM components listed above are still in the design stage. However, given the ongoing explosion of the Web technologies, one can expect the WOM approach to play a critical role in shaping the univeral persistence frameworks for the Internet and Intranets. At the moment, the single vendor models such as JDBC by Sun and OLEDB by Microsoft are ahead the consortia frameworks and in fact the Microsoft UDA solution is the most complete and advanced as of mid '98.

Sample Database Applications on the Pragmatic Object Web

In parallel with tracking the evolution of the universal data models, we are involved in building Web linked database environments for various communities and user domains. We illustrate here a few examples of such specific database applications. Practical solutions and design decisions appear often as a result of compromise between several factors such as price vs functionality tradeoffs for various technologies, project specific requirements, customer preferences etc. We indicate these tradeoffs when discussing the application examples below and then we summarize in the next section the lessons learned so far and our approach towards the HPC T&E database technology roadmap.

Careweb is a Web based collaborative environment for school nurses with support for: a) student healthcare record database; b) educational materials for nurses and parents; c) collaboration and

interactive consultation between nurses, nurse practitioners and pediatricians, including both asynchronous (shared patient records) and synchronous (audio, video, chat, whiteboard) tools. Early CareWeb prototype (Fig. 1) was developed at NPAC using Oracle7 database, PL/SQL stored procedures based programming model, and VIC/VAT collaboration tools. We found the Oracle7 model useful but hardly portable to other vendor models, especially after Oracle decided to integrate their Web and Database services.

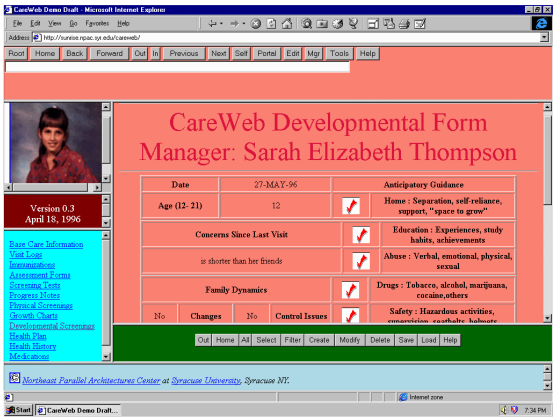

Fig 1: Sample screendump from the early CareWeb prototype at NPAC – Netscape and JavaScript with VIC / VAT collaboratory tools (video, audio, chat, whiteboard) on top of Oracle Database with PL / SQL based HTML generator.

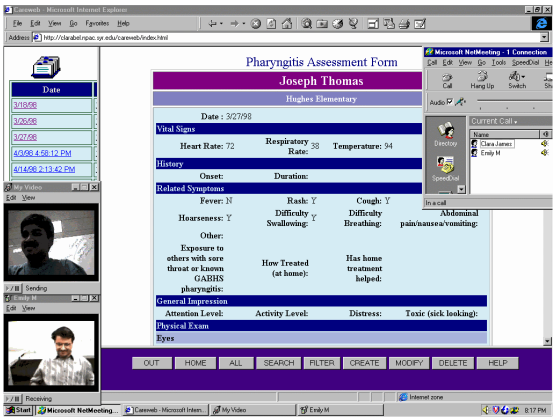

Fig 2: Sample screendump from the production CareWeb system under development by Translet, Inc. – Internet Explorer 4 with NetMeeting collaboratory (video, audio, chat, whiteboard, shared browser) on top of Access or SQL Server database with Active Server Pages based HTML

New production version of CareWeb (Fig. 2) under development by NPAC spin-off Translet, Inc. is using exclusively Microsoft Web technologies: Internet Explorer, Access/SQL Server databases, ASP/ADO based programming model, and NetMeeting based collaboration tools. Although also hardly portable beyond the Microsoft software domain, we found this solution more practical from a small business perspective as Microsoft offers now indeed a total and affordably priced solution prototype for all tiers.

Language Connect University is another Web/Oracle service constructed by Translet Inc. for the distance education community. Marketed by the Syracuse Language Systems as an Internet extension of their successful CD-ROM based multimedia courses for several foreign languages, LCU offers a spectrum of collaboratory and management tools for students and faculty of a Virtual University. Tools include customized email, student record management, interactive multimedia assignments such as quizzes, tests and final exams, extensive query services, evaluation and grading support, course management, virtual college administration and so on (see Fig. 5)

We found the Oracle based high end database solution for LCU as appropriate and satisfactory; possible follow-on projects will likely continue and extend this model by adding suitable CORBA PSS based middleware to assure compatibility with the emergent public standards for distance education such as IMS by Educom.

FMS Training Space [8][9] is an ongoing FMS PET project at NPAC within the DoD Modernization Program that develops Web based collaboratory training for FMS technologies

under development by the CHSSI program. We start with the SPEEDES training which will be gradually extended towards other FMS components such as Parallel CMS, E-ModSAF, HPC RTI, Parallel IMPORT and TEMPO/Thema. FMS Training Space combines lessons learned in our previous Web /Database projects such as CareWeb or LCU with our new WebHLA middleware based on Object Web RTI. The result will be a dynamic interactive multiuser system, with real-time synchronous simulation tools similar to online mupltiplayer gaming systems, and with the asynchronous tools for database navigation in domains such as software documentation, programming examples, virtual programming laboratory etc. Selected screendumps from the preliminary version of the FMS Training Space, including some elements of the HLA, SPEEDES, CMS and ModSAF documentation databases and demonstrated at the DoD UGC 98 Conference [8], are shown in Fig 3,4. Object Web RTI based realtime multiplayer simulation support is being included in the FMS Training Space during the summer 98 and will be demonstrated at the SIW Fall 98 Conference in Orlando [9].

Current version of the FMS Training Space is using Microsoft Web technologies: Internet Information Server, Active Server Pages, Visual Basic Script, Internet Explorer, ActiveX and Java applet plug-ins, Front Page Web Authoring and Access Database. This approach facilitates rapid prototyping in terms of the integrated web/commodity tools from Microsoft and we intend to extend it in the next stage by adding the corresponding support for UNIX, Java, Oracle and Netscape users and technologies.

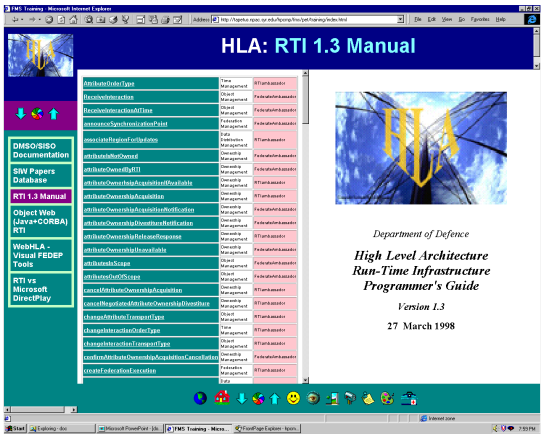

Fig 3: Sample screendump fgrom the FMS Training Space: on-line documentation database for the C++ Programming Manual for DMSO RTI 1.3

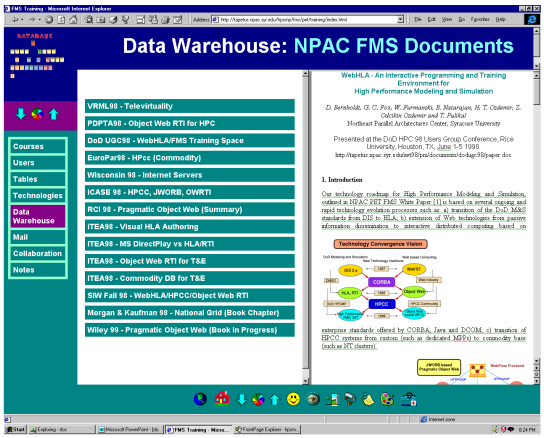

Fig 4: Sample screendump from the FMS Training Space: data warehouse component with the database of NPAC FMS '98 publications

Data Mining for Virtual Proving Ground we are working with the Virtual Proving Ground team at ARL/ATC in Aberdeen to help with the data mining services for the T&E historical test data. In particular, we received recently from the VPG a MS Access database with a vehicle engineering and testing information and we analyzed it using some selected Web and data Mining technologies.

The VPG data was made accessible over the Web using Active Server pages for ease of access across workstations and network and for use with future distributed datamining applications. An SQL Query tool was written on top of this data which can be used to run simple queries and

analyze the results. The second step was to use simple classification algorithm, C4.5, for initial Data Mining experiments. We choose one of the attributes from one the tables as our target class, dividing the values into two classes Major and Minor. We used a small subset of attributes that we thought would affect the classification process namely the subsystem, course condition and course type for our analysis. The public domain tool that we used was See5 from RuleQuest, which implements the C5.0 KDD algorithm, a successor to Quinlan's C4.5 decision tree algorithm. Training data and test data were randomly selected with the help of the query tool. The See5 tool was used to run the algorithm over the training data to generate a decision tree (see Fig. 6) and ruleset with an error rate of 3.8%. On the test cases the error rate was found to be 12%, which indicates the abnormalities in the training set selection and the decision tree generation. We are in the process of refining this to get a lower error rate and to generate a better decision tree.

Data Mining experiments as described above allow us to become familiar with the large datasets and the high performance computational challenges of T&E. In the longer run, we view VPG as a promising testbed for WebHLA based virtual prototyping environments as envisioned in Section 7.

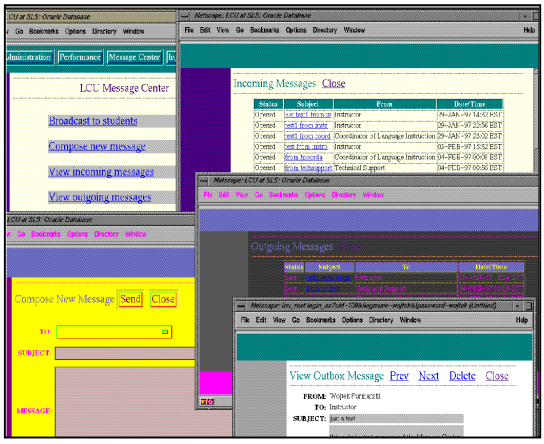

Fig 5: Sample screendump from the Language Connect University for distance education: Message Center for Collaboratory E-Mail between students and teachers

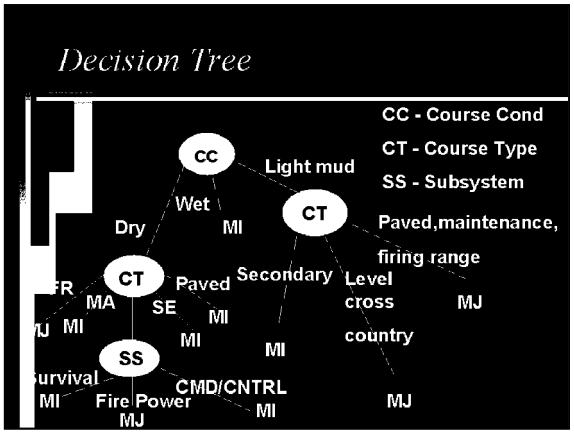

Fig 6: Decision tree generated for the Virtual Proving Ground data using the See5.0 classification algorithm

Next Steps We will continue the development of the FMS Training Space, viewed both as a practical Web linked database environment with the data warehouse of training materials such as software documentation, object model libraries, simulation scenarios etc. and as a testbed for experimenting with and evaluating new dynamic persistence technologies offered by CORBA, Java, COM and WOM. We will continue tracking the developments in the universal data technologies discussed in the first part of the paper. New successful solutions will be gradullay incorporated and made accessible within the FMS Training Space via the multi-protocol JWORB based middleware, both as enablers for higher quality professional training and as target for possible courses in the rapidly evolving domain of interactive web technologies.

7. References

1. Robert Orfali and Dan Harkey, Client/Server Programming with Java and CORBA , 2nd Edition, Wiley 1998.

2. Craig Thompson, OMG/DARPA Workshop on Compositional Software Architectures, Monterey, CA January 6-8 1998

3. G. C. Fox, W. Furmanski, H. T. Ozdemir and S. Pallickara, *Building Distributed Systems for the Pragmatic Object Web*, book in progress, Wiley '98.

4. G. C. Fox, W. Furmanski and H. T. Ozdemir, JWORB - Java Web Object Request Broker for Commodity Software based Visual Dataflow Metacomputing Programming Environment , submitted for the HPDC-7, Chicago, IL, July 28-31, 1998.

5. G. C. Fox, W. Furmanski, Subhash Nair and Z. Odcikin Ozdemir, "Microsoft DirectPlay meets DMSO RTI for Virtual Prototyping in HPC T&E Environments", to appear *in Proceedings of the International Test and Evaluation Association (ITEA) Workshop on High Performance Computing for Test and Evaluation*, Aberdeen MD, July 13-16 1998.

6. G. C. Fox, W. Furmanski and H. T. Ozdemir, "Object Web (Java/CORBA) based RTI to support Metacomputing M&S", to appear *in Proceedings of the International Test and Evaluation Association (ITEA) Workshop on High Performance Computing for Test and Evaluation*, Aberdeen MD, July 13-16 1998.

7. G. C. Fox, W. Furmanski, B. Goveas, B. Natarajan and S. Shanbhag, "WebFlow based Visual Authoring Tools for HLA Applications", to appear *in Proceedings of the International Test and Evaluation Association (ITEA) Workshop on High Performance Computing for Test and Evaluation*, Aberdeen MD, July 13-16 1998.

8. D. Bernholdt, G. C. Fox, W. Furmanski, B. Natarajan, H. T. Ozdemir, Z. Odcikin Ozdemir and T. Pulikal, WebHLA - An Interactive Programming and Training Environment for High Performance Modeling and Simulation , *in Proceedings of the DoD HPC 98 Users Group Conference*, Rice University, Houston, TX, June 1-5 1998.

9. G.C.Fox, W. Furmanski, S. Nair, H. T. Ozdemir, Z. Odcikin Ozdemir and T. Pulikal, "WebHLA - An Interactive Programming and Training Environment for High Performance Distributed FMS", to appear *in Proceedings of the Simulation Interoperability Workshop SIW Fall 98*, Orlando, FL, September 14-18, 1998.